SECURE AGGREGATION FOR PRIVACY-AWARE FEDERATED LEARNING WITH LIMITED RESOURCES

Irem Ergün*
UC Riverside
iergu001@ucr.edu

Hasin Us Sami*
UC Riverside
hsami003@ucr.edu

Başak Güler
UC Riverside
bguler@ece.ucr.edu

ABSTRACT

Secure aggregation is a popular protocol for privacy-aware model aggregation in federated learning. However, due to its large communication overhead, users with scarce wireless resources are unable to participate in the protocol as much as users with better wireless conditions, which can lead to significant bias against users from underserved communities. Towards addressing this challenge, in this work we propose a communication-efficient gradient sparsification technique for secure aggregation, where the server learns the aggregate of sparsified local gradients from a large number of users, without having access to the individual local gradients. Through large-scale distributed experiments with up to 100 users, we demonstrate significant reduction in both the communication overhead and the wall clock training time, compared to conventional secure aggregation.

1 INTRODUCTION

Federated learning is a distributed training framework to train machine learning models over the data stored at a large number of devices (users) (McMahan et al., 2017). Training is carried out in an iterative manner governed by a central server, who holds a global model. At each iteration, the server sends the current state of the global model to the users, who then update the global model through local training, generating a local model. The local models are sent from the users to the server, who then aggregates (sums) the local models to update the global model.

Due to its on-device learning architecture, federated learning is widely used in a variety of privacy-sensitive applications (Yang et al., 2019; Xu et al., 2021; Chen et al., 2020). However, the local models can still reveal privacy-sensitive information about the training data by incorporating model inversion techniques (Fredrikson et al., 2015; Nasr et al., 2019; Zhu & Han, 2020). *Secure aggregation* protocols have emerged as a countermeasure against such privacy threats, by enabling the server to *aggregate* the local models of the users, without observing the local models in the clear (Bonawitz et al., 2017; Zhao & Sun, 2021; Bell et al., 2020). In this protocol, a random mask is generated upon agreement between each pair of users, which is used to hide the actual contents of the local models before sharing them with the server.

A major challenge of secure aggregation protocols is the large communication overhead, limiting the participation of users from resource-constrained wireless environments, especially users from developing countries (Ahmad et al., 2016; Munir et al., 2021). Towards addressing this challenge, this work proposes a novel gradient sparsification technique for secure aggregation, `SparseSecAgg`, which requires only a small fraction of the local models from each user, and aggregates them without revealing their true values. Although gradient sparsification has been incorporated with conventional (non-private) learning in various notable works, (Aji & Heafield, 2017; Lin et al., 2017; Jiang & Agrawal, 2018; Stich et al., 2018; Wangni et al., 2017), it cannot be applied to secure aggregation due to its underlying cryptographic architecture. To the best of our knowledge, `SparseSecAgg` is the first gradient sparsification protocol compatible with the underlying cryptographic primitives of secure aggregation. We perform extensive experiments for image classification in a network of up to 100 users, and observe that `SparseSecAgg` significantly reduces the per-iteration communication volume compared to conventional secure aggregation. We demonstrate significant reduction in both the communication overhead and the overall (wall-clock) training time.

*Equal contribution.

2 PROBLEM FORMULATION

Federated Learning. We consider cross-device federated learning with a central server and N users, where user i has a local dataset $\mathcal{D}_i$ (McMahan et al., 2017). The goal is to train a model $\mathbf{w} \in \mathbb{R}^d$ of size d, to minimize a global loss function $F(\mathbf{w}) = \sum_{i\in[N]} \beta_i F_i(\mathbf{w})$, where F_i is the local loss function of user i and $\beta_i = \frac{|\mathcal{D}_i|}{\sum_{i\in[N]}|\mathcal{D}_i|}$ is a weight parameter. At each training round, users receive the current state of the global model $\mathbf{w}^{(t)}$ from the server, and train a local model $\mathbf{w}_i^{(t)}$ (initialized as $\mathbf{w}_i^{(t)} \leftarrow \mathbf{w}^{(t)}$) via E local SGD steps $\mathbf{w}_i^{(t)} \leftarrow \mathbf{w}_i^{(t)} - \eta^{(t)}\nabla F_i(\mathbf{w}_i^{(t)})$ where $\eta^{(t)}$ is the learning rate. User i then sends the local model $\mathbf{w}_i^{(t)}$, or the model difference $\mathbf{y}_i^{(t)} = \mathbf{w}^{(t)} - \mathbf{w}_i^{(t)}$ to the server. Finally, the server aggregates the local updates to update the global model,

$$\mathbf{w}^{(t+1)} = \sum_{i\in[N]} \beta_i \mathbf{w}_i^{(t)} = \mathbf{w}^{(t)} - \sum_{i\in[N]} \beta_i \mathbf{y}_i^{(t)} \tag{1}$$

Secure aggregation. Secure aggregation (`SecAgg`) protocol leverages secure multi-party computation principles for model aggregation in (1) without revealing the local models in the clear (Bonawitz et al., 2017). Each pair of users i, j agree on a pairwise random mask r_{ij} (unknown to others) using a cryptographic key agreement protocol. Then, each user sends a masked version of its local model,

$$\mathbf{x}_i^{(t)} = \mathbf{w}_i^{(t)} + \sum_{j:i<j} \mathbf{r}_{ij}^{(t)} - \sum_{j:i>j} \mathbf{r}_{ij}^{(t)} \tag{2}$$

to the server. The pairwise masks hide the contents of the local models from the server (in an information-theoretic sense). On the other hand, when the server aggregates the masked models, the pairwise masks $r_{ij}^{(t)}$ cancel out, and the server learns the aggregate of true local models $\sum_{i\in[N]} \mathbf{w}_i^{(t)}$.

The communication overhead of sending large local models poses a major limitation for resource-constrained users. Gradient sparsification is a popular approach to address this challenge in (non-private) federated learning (Aji & Heafield, 2017; Jiang & Agrawal, 2018; Stich et al., 2018; Wangni et al., 2017). The most common techniques are rand-K and top-K sparsification, where each user sends only $K \ll d$ gradient parameters to the server, selected randomly (rand-K) or with respect to their magnitude (top-K). On the other hand, the locations of the K parameters often vary from one user to another, hence these techniques cannot be applied to secure aggregation, as the random masks in (1) will not cancel out upon aggregation. Our goal is to address this challenge, by developing a gradient sparsification framework for secure-aggregation, where the server learns the aggregate of *sparsified local gradients* from a large number of users, but without learning the local parameters.

Threat Model. Similar to conventional secure aggregation works, our focus is on an honest-but-curious adversary model, where adversarial parties follow the protocol truthfully, but try to extract the local data samples from the information exchanged during training. Out of N users, up to $A \leq \gamma N$ users are adversarial for some $\gamma \in (0, 0.5)$, who may collude with each other and/or the server.

Key performance metrics. We evaluate the performance of `SparseSecAgg` according to the following key performance metrics: 1) *Aggregation cardinality:* The aggregation cardinality, T, quantifies the number of honest users participating at any given location of the global model. In the context of secure aggregation, a larger T provides better obfuscation. 2) *Compression ratio:* The compression ratio $\alpha \in (0, 1]$ is defined as the fraction of the entire gradient vector each user sends to the server (a smaller α reduces the communication overhead).

3 THE SPARSESECAGG FRAMEWORK

Generation of Pairwise additive masks. `SparseSecAgg` uses additive masks to hide the true value of the local model updates from the server during aggregation. Towards that end, each pair of users $i, j \in [N]$ agree on a pairwise random seed s_{ij} via leveraging Diffie-Hellman key exchange protocol (Diffie & Hellman, 1976). Users $i,\ j$ then expand the random seed s_{ij} using a PRG (pseudorandom number generator), to a vector $\mathbf{r}_{ij} = \text{PRG}(s_{ij}) \in \mathbb{F}_q^d$, where each element is generated uniformly at random from a finite field $\mathbb{F}_q$ of integers modulo a prime q.

Generation of Pairwise binary masks. To determine the sparsification pattern, each pair of users $i, j \in [N]$ agree on a random binary vector $\mathbf{b}_{ij} \in \{0, 1\}^d$, where each element $\ell \in [d]$ is drawn

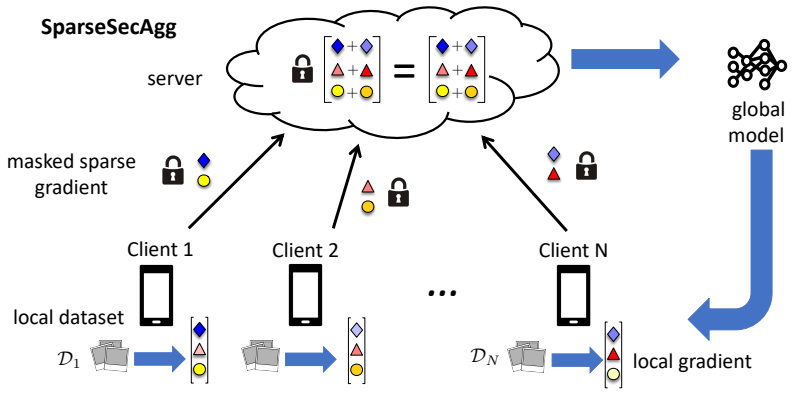


Figure 1: The SparseSecAgg framework.

from an IID Bernoulli distribution with success probability $\frac{\alpha}{N-1}$ for $\alpha \in (0, 1]$, which represents the compression ratio. To generate the binary vectors, firstly, another instantiation of the process described above for additive mask generation is carried out, where a vector of size d is generated uniformly at random from $\mathbb{F}_q$. Then, the domain of the PRG is divided into two intervals, where the size of the intervals are proportional to $\frac{\alpha}{N-1}$ and $1 - \frac{\alpha}{N-1}$, respectively. The numbers in the first interval are mapped to 1 whereas the others are mapped to 0. By doing so, each pair of users i, j agree on $\mathbf{b}_{ij} = \mathbf{b}_{ji} \in \{0, 1\}^d$. These masks denote the coordinates of the parameters which are selected by the users to be sent to the server, and ensure that the additive masks cancel out upon aggregation.

Sparsified Quantized Gradient Construction. User $i \in [N]$ scales and quantizes their local gradient $\mathbf{y}_i$ via stochastic quantization, and computes the scaled quantized gradient $\overline{\mathbf{y}}_i$. User i then forms a sparsified masked gradient $\mathbf{x}_i$, where,

$$\mathbf{x}_i(\ell) = \Big(1 - \prod_{j \in [N]: j \neq i} (1 - \mathbf{b}_{ij}(\ell))\Big)(\overline{\mathbf{y}}_i(\ell)) + \sum_{j \in [N]: i < j} \mathbf{b}_{ij}(\ell)\mathbf{r}_{ij}(\ell) - \sum_{j \in [N]: i > j} \mathbf{b}_{ij}(\ell)\mathbf{r}_{ij}(\ell) \quad (3)$$

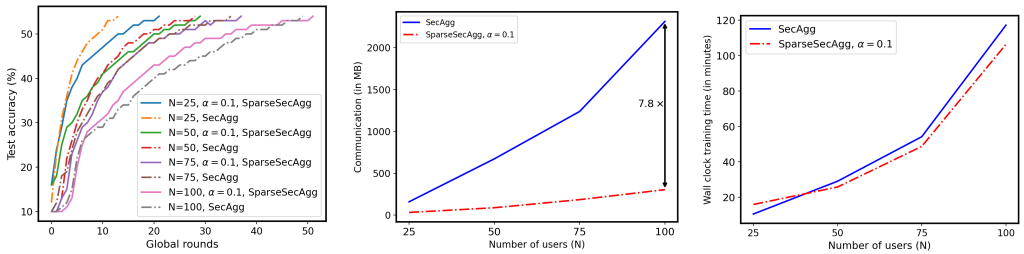
for $\ell \in [d]$. Specifically, for each non-zero element $\mathbf{b}_{ij}(\ell)$, user i subtracts $\mathbf{r}_{ij}(\ell)$ from its quantized gradient $\overline{\mathbf{y}}_i(\ell)$ if $i > j$, and adds it if $i < j$, which ensures cancellation of pairwise additive masks upon aggregation at the server. Next, we define a set $\mathcal{U}_i$ that contains the coordinates of the gradient parameters selected by user i. User i then sends all $\mathbf{x}_i(\ell)$ for $\ell \in \mathcal{U}_i$, along with $\mathcal{U}_i$ to the server. This process results in only αd gradient parameters being shared with the server on average, which substantially improves the per-round communication efficiency as we demonstrate in our experiments. Moreover, communicating sparse updates has the added benefit of being more resilient against model inversion attacks (Sun et al., 2021; Zhao et al., 2020).

Secure Aggregation of Sparsified Gradients. The server aggregates (sums) the sparsified masked gradients $\mathbf{x}_i$ received from the surviving users. Upon aggregation, the pairwise additive masks $\mathbf{r}_{ij}$ cancel out from the sum $\sum_i \mathbf{x}_i$, allowing the server to learn the sum of the true sparsified gradients $\sum_i \overline{\mathbf{y}}_i$, without observing the true content of each gradient. Handling of dropout users is carried out as in standard secure aggregation (Bonawitz et al., 2017). The sum of the quantized gradients, $\overline{\mathbf{y}}(\ell)$ are then de-quantized from the finite field to the real domain to update the global model, which is then sent to the users for the next round. The SparseSecAgg framework is illustrated in Fig. 1.

4 Experiments

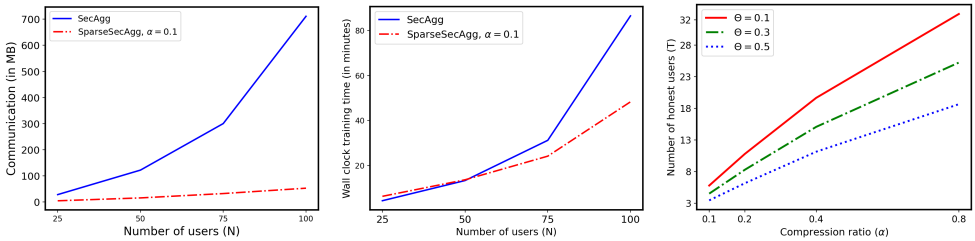
We consider image classification tasks on CIFAR-10 (Krizhevsky & Hinton, 2009) and MNIST (LeCun et al., 2010) datasets, with the CNN architectures from McMahan et al. (2017). We consider both IID and non-IID data distributions for the users from McMahan et al. (2017) for $N = 25,\ 50,\ 75,\ 100$. The compression ratios are $\alpha = 0.1,\ 0.2$, and the dropout rate is set to $\theta = 0.3$. We run all experiments on Amazon EC2 *m4.large* machine instances. The bandwidth of the users are set to 100 Mbps to emulate bandwidth-limited mobile environments. We use mini-batch gradient descent with batch size 28, momentum 0.5 and learning rate 0.01. We set the number of local training epochs (E) to 5.

In Fig. 2(a), we demonstrate the convergence behavior of SparseSecAgg with $\alpha = 0.1$ versus the conventional secure aggregation protocol from Bonawitz et al. (2017) (termed SecAgg). In Fig.s



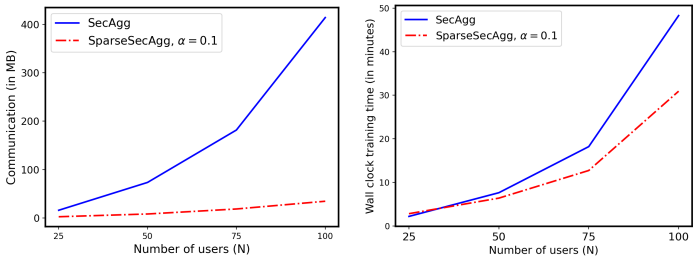
(a) Test accuracy vs global rounds. (b) Total communication overhead. (c) Wall clock training time.

Figure 2: Performance comparisons on the CIFAR-10 dataset distributed IID across users, with the CNN architecture from Bonawitz et al. (2017) for a target accuracy 53%.



(a) Total communication overhead. (b) Wall clock training time. (c) Aggregation cardinality (T) for $N = 100$, $\gamma = 0.3$.

Figure 3: Results on the MNIST dataset distributed IID across users, with the target test accuracy set to 97% (a, b) and aggregation cardinality of `SparseSecAgg` (c).



(a) Total communication overhead. (b) Wall clock training time.

Figure 4: Results on the MNIST dataset distributed non-IID across users (with target accuracy 94%).

2(b), 3(a), and 4(a), we demonstrate the total communication overhead to reach a target accuracy for `SparseSecAgg` and `SecAgg`. We observe that for CIFAR-10, `SparseSecAgg` reduces the communication overhead by up to $7.8\times$. In addition, `SparseSecAgg` also speeds up the wall-clock training time by up to $1.13\times$ compared to `SecAgg`, as demonstrated in Fig. 2(c). We also note that per-round communication overhead is reduced by around $8\times$ for $\alpha = 0.1$. In Fig. 3(c), we fix $N = 100$, $\gamma = 0.3$ and show the linear trade-off between aggregation cardinality (T) and the compression ratio (α) for various dropout rates.

5 Conclusion

We propose a sparsified secure aggregation framework, `SparseSecAgg`, for federated learning under scarce wireless resources. `SparseSecAgg` enables aggregation of sparsified gradients from a large number of users, without revealing them in the clear. Our experiments demonstrate a significant improvement in the communication overhead and wall-clock training time.

6 Acknowledgements

Irem Ergün would like to thank Ömer Eren for his contribution to the codebase and helpful discussions. This material is based in part on work supported by the OUSD (R&E) / RT&L Cooperative Agreement W911NF-20-2-0267, NSF CAREER grant CCF-2144927, and a UC Regents Faculty Award. The views, opinions, or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of ARL and OUSD (R&E) / RT&L or the U.S. Government.

REFERENCES

Sohaib Ahmad, Abdul Lateef Haamid, Zafar Ayyub Qazi, Zhenyu Zhou, Theophilus Benson, and Ihsan Ayyub Qazi. A view from the other side: Understanding mobile phone characteristics in the developing world. In *Proceedings of the 2016 Internet Measurement Conference*, pp. 319–325, 2016.

Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021*, 2017.

James Henry Bell, Kallista A Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly) logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1253–1269, 2020.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93, 2020.

Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333, 2015.

Peng Jiang and Gagan Agrawal. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 2530–2541, 2018.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *http://yann. lecun. com/exdb/mnist*, 2010.

Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, Fort Lauderdale, FL, USA, Apr 2017.

Muhammad Tahir Munir, Muhammad Mustansar Saeed, Mahad Ali, Zafar Ayyub Qazi, and Ihsan Ayyub Qazi. Fedprune: Towards inclusive federated learning. *arXiv preprint arXiv:2110.14205*, 2021.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753. IEEE, 2019.

Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2018.

Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9311–9319, 2021.

Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. *arXiv preprint arXiv:1710.09854*, 2017.

Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

Yizhou Zhao and Hua Sun. Information theoretic secure aggregation with user dropouts. In *IEEE International Symposium on Information Theory, ISIT'21*, 2021.

Ligeng Zhu and Song Han. Deep leakage from gradients. In *Federated Learning*, pp. 17–31. Springer, 2020.